Trustworthy Assurance

Leveraging Institutional Knowledge in Software Development

Jerry Cochran
Sr. Security Strategist
Advanced Strategies & Policy/TwC

Agenda

- Session Goals
- Overview and Background
- The SDLC: State of the Art vs. Vision
- Institutional Knowledge & Assurance
- Discussion & Feedback

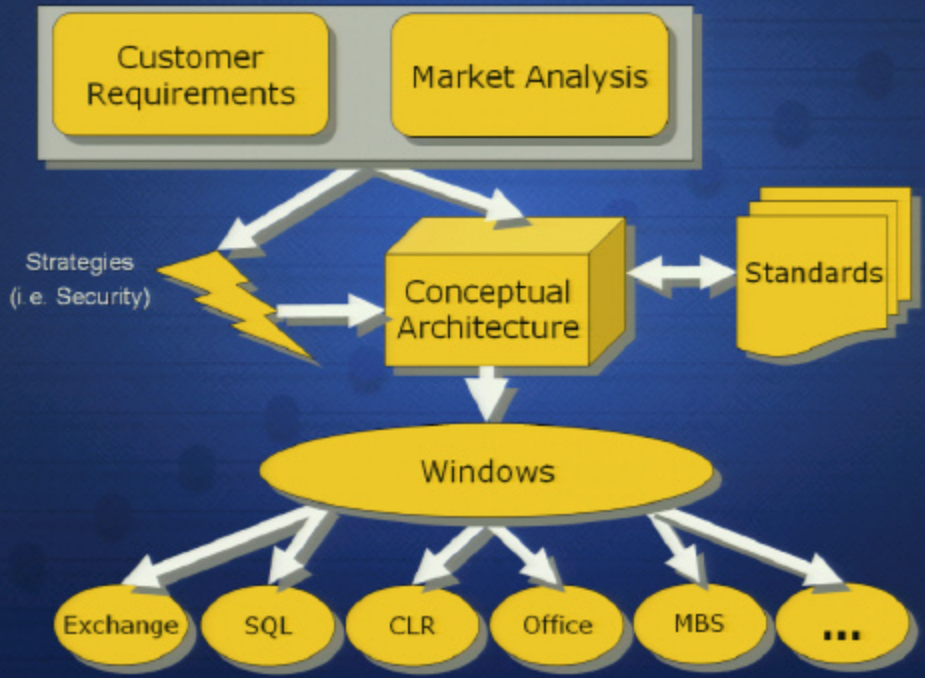
Session Goals

- Share a vision of the SDLC
- Engage Your Expertise
- Exchange Ideas and Solicit Your Feedback

Overview and Background

- Trustworthy Computing has changed Microsoft
  - Security Development Lifecycle & Engineering Excellence
  - Architecture-driven development
- What About…
  - Institutional Knowledge opportunity
  - Measuring "trustworthiness"
  - Error Prevention vs. Error Detection
  - Architecture compliance/assurance

Architectural Knowledge



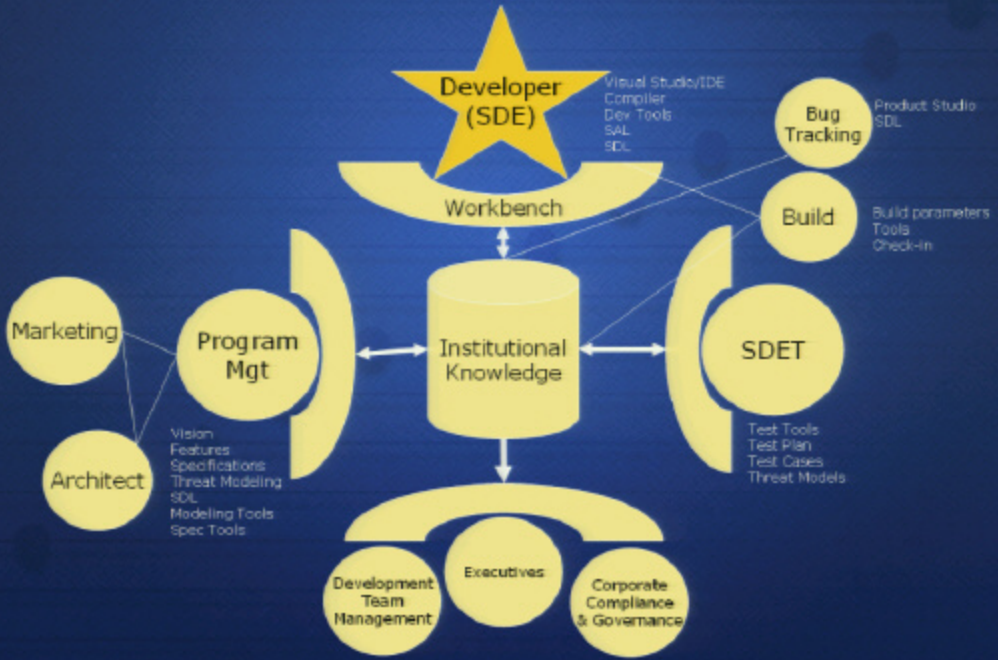
SDLC: State of the Art vs. Vision

	Today	Future
Requirements	Human & documents	Human & metadata
Specification	Documents (sometimes)	Docs & tools with metadata
Threat Modeling	Manual, document-based	Metadata-driven docs & tools
Coding & Debugging	Developer preference and practice ("visual notepad")	Metadata-driven Unified IDE
Code/Object Reuse	Partial and manual	Complete and automated
Test Development	Manual & feature-based	Automated and knowledge-based
Error Detection	Mostly Manual and subjective	Automated and knowledge-based
Error Prevention	Training & Education only	Various automated mechanisms throughout SDLC
Architectural Compliance	None or manual review	Automated
Institutional Knowledge	Oral tradition or chance	Captured as metadata

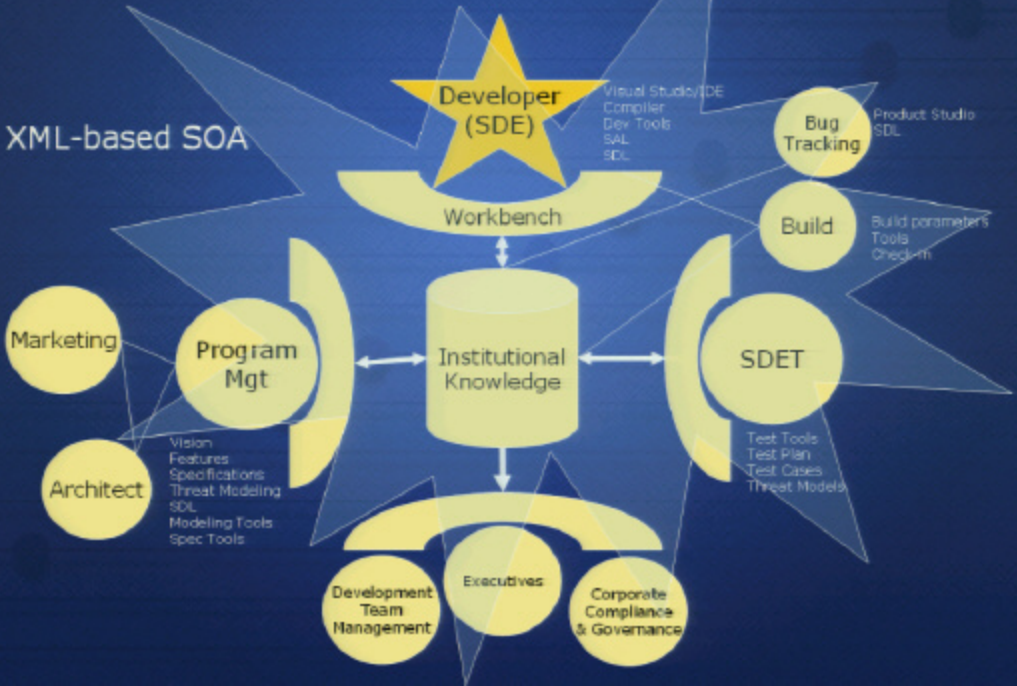
Institutional Knowledge is the Key

- All Sources
  - Architecture, Business Rules, Best Practices, etc.
- Must be "Consumable"
  - Machine and Human Readable Metadata
- Used Throughout the SDLC
  - By Processes, Workflow, and Tools
- Benefits
  - A shift to error prevention (in addition to detection)
  - Measurement and Assurance
  - Institutional Knowledge is captured and utilized
  - Improved quality, efficiency, and trustworthiness

Knowledge and Assurance



Knowledge and Assurance



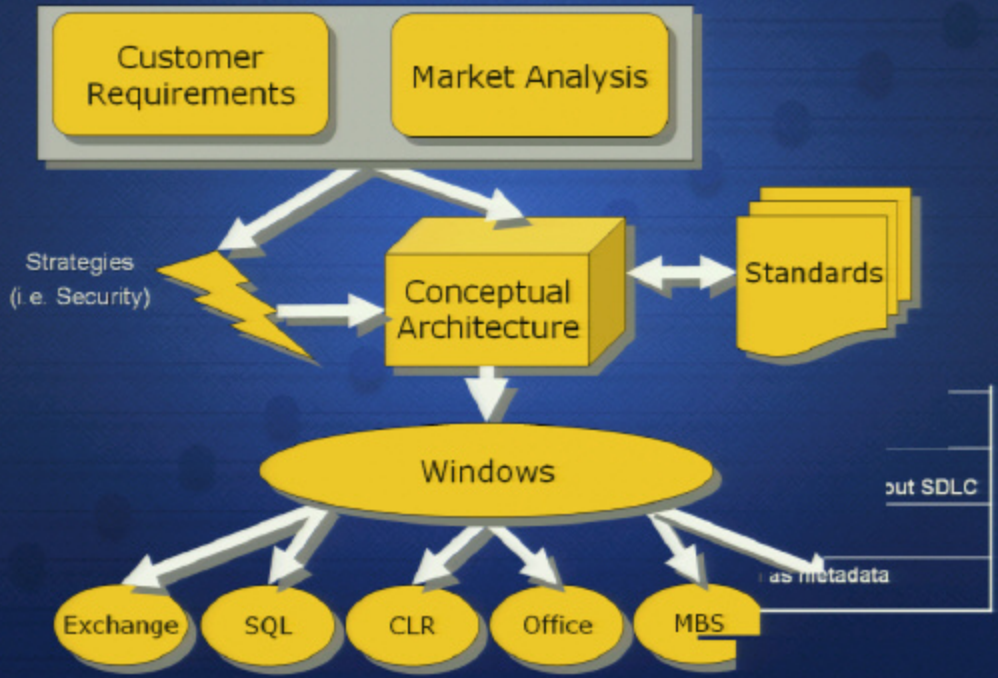
Questions and Discussion

- Feasibility
  - What components of the vision are realistic and which are not?
- Technical Challenges
  - Does the "State of the Art" in architecture-driven development support the vision?
  - What technologies exist to make knowledge consumable in this fashion?
  - What SDLC tools are needed?
- Business and Cultural Issues?
- What else/Other thoughts?

Institutional Knowledge is the Key

- All Sources
  - Architecture, Business Rules, Best Practices, etc.
- Must be "Consumable"
  - Machine and Human Readable Metadata
- Used Throughout the SDLC
  - By Processes, Workflow, and Tools
- Benefits
  - A shift to error prevention (in addition to detection)
  - Measurement and Assurance
  - Institutional Knowledge is captured and utilized
  - Improved quality, efficiency, and trustworthiness

Architectural Knowledge



Overview and Background

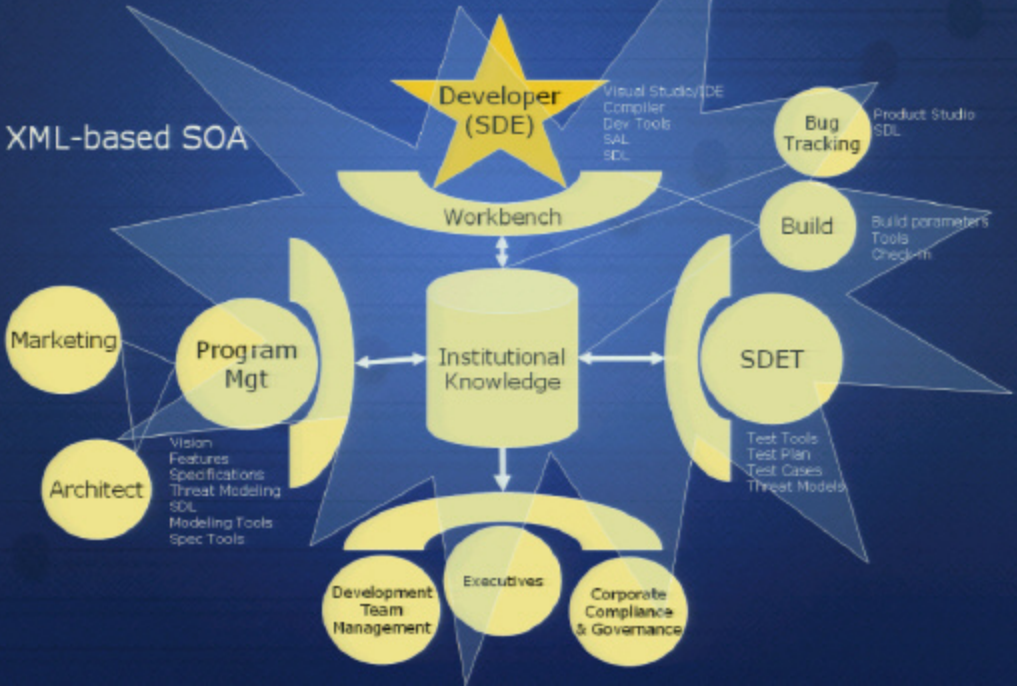
- Trustworthy Computing has changed Microsoft
  - Security Development Lifecycle & Engineering Excellence
  - Architecture-driven development
- What About…
  - Institutional Knowledge opportunity
  - Measuring "trustworthiness"
  - Error Prevention vs. Error Detection
  - Architecture compliance/assurance

Institutional Knowledge vs. Vision the Key

- All Sources
  - Architecture, Bu
- Must be "Cor
  - Machine an
- Used Thr
  - By Pr
- Bene
  - A
  - revention
  - rchitectural Compliance
  - nstitutional Knowledge

	Future
documents	Human & metadata
cuments (sometimes)	Docs & tools with metadata
nual, document-based	Metadata-driven docs & tools
Developer preference and practice ("visual notepad")	Metadata-driven Unified IDE
Partial and manual	Complete and automated
Manual & feature-based	Automated and knowledge-based
Mostly Manual and subjective	Automated and knowledge-based
Training & Education only	Various automated mechanisms throughout SDLC
None or manual review	Automated
Oral tradition or chance	Captured as metadata

Knowledge and Assurance



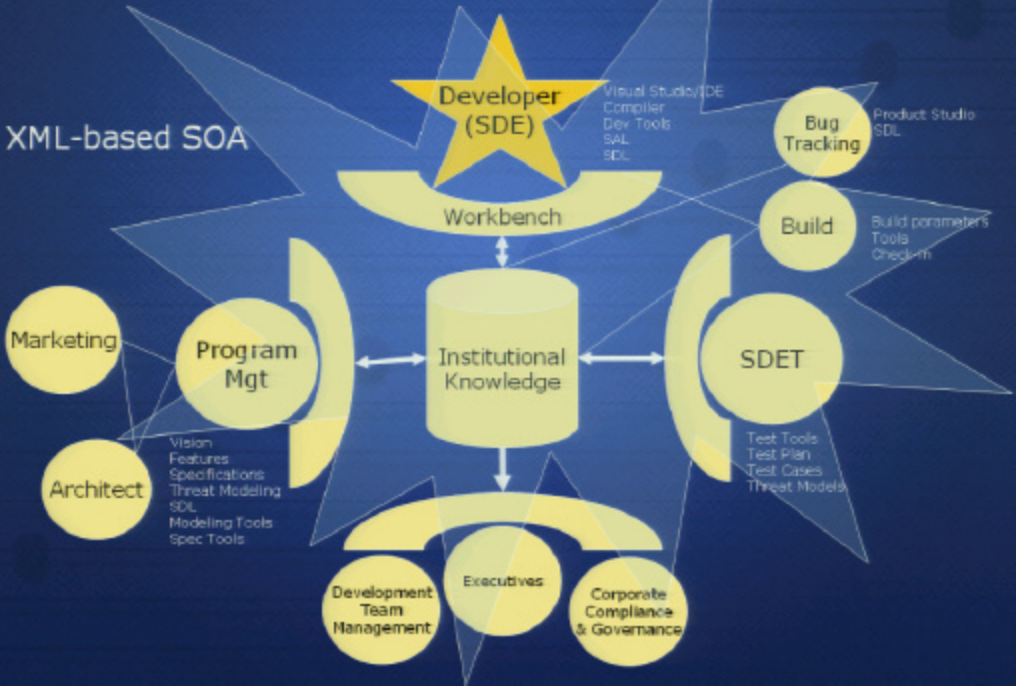
Questions and Discussion

- Feasibility
  - What components of the vision are realistic and which are not?
- Technical Challenges
  - Does the "State of the Art" in architecture-driven development support the vision?
  - What technologies exist to make knowledge consumable in this fashion?
  - What SDLC tools are needed?
- Business and Cultural Issues?
- What else/Other thoughts?

Questions and Discussion

- Feasibility
  - What components of the vision are realistic and which are not?
- Technical Challenges
  - Does the "State of the Art" in architecture-driven development supp the vision?
  - What technologies exist to make knowledge consumable in this fo
  - What SDLC tools are needed?
- Business and Cultural Issu
- What else/Other thoug

Knowledge and Assurance



Microsoft®

Your potential. Our passion.™

Feedback Please!

jerryco@microsoft.com